

# Лекція №9. Структури

---

ПРОГРАМУВАННЯ ТА ПРИКЛАДНІ ІНФОРМАЦІЙНІ  
СИСТЕМИ

# Визначення

---

**Структура** — це сукупність різнотипних елементів, яким присвоюється одне ім'я (воно може бути відсутнім), що займає одну ділянку пам'яті. Елементи, що складають структуру, називаються *полями*.

Змінна типу структура, як і будь-яка змінна, повинна бути описана. Цей опис складається з двох кроків: опису шаблону (тобто складу) або типу структури та опису змінних структурного типу.

# Опис структури

---

```
struct <назва типу структури>
```

```
{
```

```
  <тип 1><назва поля 1>;
```

```
  ...
```

```
  <тип 2><назва поля 2>;
```

```
};
```

де **struct** — службове слово;

< назва типу структури > — ім'я типу структура (може бути відсутнім);

<тип 1>, <тип 2> — імена стандартних або визначених типів;

ім'я поля 1, ім'я поля 2,... — імена полів структури;

# Приклад 1

---

```
struct grupa
```

```
{
```

```
    char name[20];
```

```
    char surname[20];
```

```
    float srba;
```

```
};
```

# Змінні типу структура

---

Коли в програмі описана структура, то оголосити змінні або вказівники цього типу можна так:

```
grupa gr1, gr2;
```

```
grupa *p;
```

# Змінні типу структура

---

Оголосити змінні типу структура можна ще й так:

```
struct <назва типу структури>
```

```
{
```

```
    <тип поля 1><назва поля 1>;
```

```
    ...
```

```
    <тип поля n><назва поля n>;
```

```
} <список змінних і вказівників>;
```

# Приклад 2

---

```
struct grupa
```

```
{
```

```
    char name[20];
```

```
    char surname[20];
```

```
    float srba;
```

```
} gr1, gr2, *p;
```

# Структура може містити поля типу структура.

---

Приклад 3. Оголосимо структуру `grupa` та дві змінні `gr1`, `gr2` цього типу. Нехай структура `grupa` містить поля прізвище, ім'я, середній бал і структуру дати народження (`birthday`), яка складається з полів: рік, місяць, число.



# Приклад 3

---

```
struct grupa {  
    char name[20];  
    char surname[20];  
    struct birthday {  
        int year, month, day;  
    } btday;  
    float srba1;  
} gr1, gr2;
```

# Ініціалізація змінних

---

Змінні типу структура можна ініціалізувати відразу під час оголошення або звернувшись то створеного типу.

```
struct grupa {  
    char name[20];  
    char surname[20];  
    float srbal;  
} gr1 = { "Олександра", "Ярмолюк", 4.9 };
```

```
grupa gr2 = { "Ivan", "Ivanov", 2.3 };
```

# Доступ до полів

---

Доступ до конкретного поля змінної типу структура дає складене ім'я вигляду:

**<назва змінної>.<назва поля>**

Можна також створювати вказівники на структури. Доступ до полів вказівника на структуру здійснюється дещо інакше, ніж до полів відповідної змінної, а саме:

**<назва вказівника> -> <назва поля>**

# Приклад

---

Оголосимо вказівник `p` на структуру `grupa` з прикладу 3 та проініціалізуємо його.

```
struct grupa {  
    char name[20];  
    char surname[20];  
    struct birthday {  
        int year, month, day;  
    } btday;  
    float srba1;  
} gr1, gr2;
```

```
grupa *p;  
p = &gr1;  
p->srba1 = 4.7;  
p->btday.year = 1986;
```

# Масив змінних типу структура

---

Масив змінних типу структура можна оголосити так:

**<назва типу структура> <назва масиву >  
[кількість елементів];**

Доступ до полів окремих елементів масиву здійснюється так:

**<назва масиву>[індекс].<назва поля>**

# Структури і функції

---

Змінні типу структура можуть бути аргументами функцій. Для них діють ті ж самі правила, що і для змінних стандартних типів.

# Задача 1

---

Використовуючи масив типу структура, скласти програму, за допомогою якої можна ввести у пам'ять комп'ютера дані про студентів групи ПП-11 (ім'я, прізвище, дата народження, середній бал) та вивести окремо студентів з середнім балом менше ніж 3.

---

```
#define n 3
struct grupa{
    char name[20];
    char surname[20];
    struct{
        int year, month, day;
    }birthday;
    float srbal;
};
int i;
void stud_bad(struct grupa *p);
```



```

void main(){
grupa kn2[n];
for (i = 0; i<n; i++){
    cout << "Vvedit imya ta familiu " << i + 1 << "-go studenta\n";
    cin >> kn2[i].name >> kn2[i].surname;
    cout << "\nVvedit datu naridzennya den-misaz-rik\n";
    cin >> kn2[i].birthday.day >> kn2[i].birthday.month >>
        kn2[i].birthday.year;
    cout << "\nVvedit serednij bal studenta\n";
    cin >> kn2[i].srbal;
    cout << "\n-----\n";
}
cout << "Studenti grupi kn2:\n";
for (i = 0; i<n; i++){
    cout << kn2[i].name << "\t" << kn2[i].surname << "\t" <<
kn2[i].birthday.day << "." << kn2[i].birthday.month << "." <<
kn2[i].birthday.year;
    cout << "\t" << kn2[i].srbal << "\n";
}
stud_bad(kn2);

system("pause"); // console pause
}

```

---

```
void stud_bad(struct grupa *p){
    cout << "\n-----\n";
    cout << "\nStudenti, yaki pogano v4atsya\n";

    for (i = 0; i<n; i++){
        if (p[i].srbal<3){
            cout << p[i].name << "\t" << p[i].surname << "\t" <<
                p[i].birthday.day << "." << p[i].birthday.month <<
                "." << p[i].birthday.year;
            cout << "\t" << p[i].srbal << "\n";
        }
    }
}
```